



UNITED STATES PATENT AND TRADEMARK OFFICE

NN

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/614,396	07/07/2003	Lin Yan	10016253-1	5944
22879	7590	07/18/2007	EXAMINER	
HEWLETT PACKARD COMPANY			ZHE, MENG YAO	
P O BOX 272400, 3404 E. HARMONY ROAD			ART UNIT	PAPER NUMBER
INTELLECTUAL PROPERTY ADMINISTRATION				
FORT COLLINS, CO 80527-2400			2109	
MAIL DATE		DELIVERY MODE		
07/18/2007		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/614,396	YAN ET AL.
	Examiner	Art Unit
	MengYao Zhe	2109

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 18 April 2007.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1 to 34 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1 to 34 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO/SB/08)
 Paper No(s)/Mail Date _____.
 4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____.
 5) Notice of Informal Patent Application
 6) Other: _____.

DETAILED ACTION

1. Claims 1-34 are presented for examination.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 1-34 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

- a. The following claim languages are unclear and indefinite:

- i) As per claim 1, line 1, it is unclear how the module type definition table is maintained since nothing is ever done to it throughout this claim < i.e. does the table get changed after detecting an undefined module? > line 3, it is uncertain what is meant by identifying < i.e. is there an ID or name associated with the module type? >. Line 4, it is uncertain what is meant by "external module type definition table" < i.e. what is the table external to? The operating system? >. It is also unclear if "a module type definition table" found in line 1 refers to the "external module type definition table" <

i.e. are they the same definition table? >. Line 5, it is uncertain how the method is able to determine that the module type is not defined < i.e. is there a list of IDs in the table that the invention goes through to see if the ID of the module type cannot be found in the list in the table? >. Line 6, it is unclear what is meant by dynamically creating a definition < i.e. is a random ID automatically assigned to the module? What is the definition, an ID associated with the module? >. Line 8, it is unclear what is meant by “at the direction” of the static operating system kernel. It is phrased confusingly.

- ii) As per claim 2, line 2, and similarly claim 12, it is unclear what is meant by “operator generated” < i.e. does the operator have to be human? >.
- iii) As per claims 2, 3, 12, 13, it is unclear what is meant by “a DLKM type identifier”.
- iv) As per claims 4, 14 , 24, it is unclear what is meant to “conduct pre-registration support”
- v) As per claims 5, 15, it is unclear what is meant to “conduct registration function”
- vi) As per claims 6, 16, 25, it is unclear what is meant to “conduct post-registration support”
- vii) As per claims 7, 17, 22, it is unclear what is meant to “conduct pre-loading support”

viii) As per claim 8, 18, 23, it is unclear what is meant to "conduct post-loading support"

viiii) As per claim 11, line 1, it is unclear how the module type definition table is maintained since nothing is ever done to it throughout this claim < i.e. does the table get changed after detecting an undefined module? > Line 3, it is unclear how the logic can detect a module is undefined < i.e. does it look through the table? >. Line 7, it is unclear how the logic can identify a support module associated with the module < i.e. is there a list of support module IDs that is in the module, from which the system have to look at to find the support modules? >. Line 11-12, it is unclear what is meant by "externally storing data defining the module type" < i.e. external to the operating system? > It is also unclear how the data defining the module type is related to the definition table and the new module type < i.e. do the data correspond to the new module type? Is it stored in the definition table? >. Lines 5, 7, 9, 11 all recites "a computer readable medium", it is unclear as to how they are related < i.e. are they the same computer readable medium? >.

x) As per claim 21, line 2, it is unclear how the module type definition table is maintained since nothing is ever done to it throughout this claim. Line 4, it is uncertain what is meant by identifying. Line 5, it is unclear how the instructions can determined a module is undefined < i.e. not in found in the

table? >. Line 6, it is unclear how data relates to module type <i.e. do the data contain module type information? >

xi) As per claim 26, line 1, it is unclear what a static operating kernel is. Line 4, it is unclear how identifying is done <i.e. does it look at an ID? >. Line 5, it is unclear what is meant by an “external module type definition table” <i.e. what is it external to? > Lines 3, 4, 6, 8, 7, 9, 11, all recites “a computer readable medium”, it is unclear as to how they are related <i.e. are they the same computer readable medium? >.

xii) As per claims 28 and 29, both claim for an operator. It is unclear what an operator might be <i.e. does it have to be human? >.

xiii) As per claim 30, line 1, it is unclear what a static operating system kernel is <i.e. is the operating system kernel not executing anything? >. Line 6, it is unclear what an “external module type reference table” is <i.e. what is the table external to? > Through out claim 30 and its dependent claims 32, 33, and 34, they all recites “a computer readable medium”, it is unclear as to how they are related <i.e. are they the same computer readable medium? >.

xiv) As per claim 32, it is unclear what an operator might be.

xv) As per claim 33, line 4, it is unclear what “software generated module type” is. <i.e. how can a software generate a module? Is the software calling for a new function to be loaded into the system? Or is it the case that the software is the module itself? >

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1, 9, 10, 30, 31, and 34 are rejected under 35 U.S.C. 102(e) as being anticipated by Berg et al., Patent No. 6449660 (Hereafter Berg).

6. As per claim 1, Berg teaches a method for maintaining a module type definition table (*Column 16, lines 55- 60; Column 14, lines 8-21; Column 14, lines 43-60*) by a statically configured portion of an operating system kernel, comprising:

Identifying a module type (class information is the module type) (*Column 21, lines 15-33*)

Searching an external module type definition table for the module type (*Column 16, lines 55-60; Column 14, lines 8- 21; Column 14, lines 43-60*)

Determining the module type is not defined in the module type definition table (*Column 21, lines 32-34*)

Dynamically creating a module type definition (*Column 21, lines 28- 41*);
Updating an external module type definition table to include the module type definition at the direction of the static operating system kernel (*Column 21, lines 28-40*).

7. As per claim 9, Berg teaches:

Wherein creating a module type definition includes receiving at least one of a pointer and a reference, each at least one of a pointer and a reference being respectively associated with a support module (*Column 14, lines 49-51; Column 17, lines 27-40: the child device corresponds to the support module.*)

8. As per claim 10, Berg teaches:

creating a module type definition includes receiving at least one symbol name, each symbol name being respectively associated with a support module (*Column 14, lines 43-50; Column 17, lines 25- 40*)

9. As per claims 30 and 31, it has all the logics that are also contained in claim 1. Since claim 1 is rejected, claims 30 and 31 are rejected as well.

10. As per claim 34, Berg teaches logic on a computer readable medium to identify at least one support module associated with the module type. (Column 14, lines 44-59)

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

12. Claims 4-8, 11 and 14-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Berg et al., Patent No. 6449660 (Hereafter Berg) in view of: The Windows NT Device Driver Book by Art Baker, 1997 (hereafter Baker).

13. As per claim 11, Berg teaches a system for maintaining a module type definition table (*Column 14, lines 13-16*), comprising:

module type detection logic on a computer readable medium for detecting that a module is of an undefined module type; (*Column 15, lines 5-24*;

Column 21, lines 32-34: the names used to identify the device including Rtok, RscName correspond to module type.)

module type identification logic on a computer readable medium for assigning a new module type associated with the module; (*Column 21,*

lines 32-34)

support module identification logic on a computer readable medium for identifying at least one support module associated with the module; (*Column 14, lines 45- 53; Column 17, lines 24-40: the child device corresponds to the support module.*)

module type definition logic on a computer readable medium for externally storing data defining the module type. (*Fig 8, unit 813 is external to the operating system.*)

14. Berg does not teach support module loading logic on a computer readable medium for loading the at least one identified support module. However, Baker teaches loading one identified support module (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to modules, that need to be loaded, the third driver may specify all the modules it depends on, which corresponds to the support modules, for them to be loaded.)

15. It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to have modified the invention of Berg, with: support module loading logic on a computer readable medium for loading the at least one identified support module, because it allows the user to created load-order dependency.

16. The Examiner notes that the system as claimed in 11 has the capabilities to do all of the method steps found in claim 1.

17. As per claim 14, Baker teaches a support module operative to conduct pre-registration support. (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to modules, that need to be loaded, the first driver that needs to be loaded before the second driver corresponds to the pre-registration support module. The Examiner considers the pre-registration support module to be the equivalent of pre-loading module)
18. As per claim 15, Baker teaches a module to conduct a registration function (Pg 414, *third bullet: in the case that the second module is dependent on the first module, the DependOnGroup specifies names of modules that the second module depends on.* This corresponds to registration.);
19. As per claim 16, Baker teaches a post-registration module (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to modules, that need to be loaded, the third driver that needs to be load after the second driver is considered to be the post-registration module.)
20. As per claim 17, Baker teaches a support module operative to conduct pre-loading support. (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to modules, that need to be loaded, the first driver

that needs to be loaded before the second driver corresponds to the pre-loading support module.)

21 As per claim 18, Baker teaches a post-loading module (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to modules, that need to be loaded, the third driver that needs to be load after the second driver is considered to be the post-loading module. The Examiner considers the post-registration support module to be the equivalent of post-loading support.)

22. As per claim 19, Berg teaches:

Wherein creating a module type definition includes receiving at least one of a pointer and a reference, each at least one of a pointer and a reference being respectively associated with a support module (*Column 14, lines 49-51; Column 17, lines 27-40: the child device corresponds to the support module.*)

23. As per claim 20, Berg teaches:

creating a module type definition includes receiving at least one symbol name, each symbol name being respectively associated with a support module (*Column 14, lines 43-50; Column 17, lines 25- 40*)

24. As per claim 21, it contains all the instructions necessary to perform the method steps capable by the system of claim 11. Since claim 11 is rejected, claim 21 is rejected as well.

25. As per claim 26, Berg teaches a static operating kernel on a computer readable medium (Fig 8) comprising:

logic on a computer readable medium to receive a request to load a module
(Column 14, lines 55-60)

logic on a computer readable medium to identify a module type of the module
(Column 21, lines 15-26)

logic on a computer readable medium to reference an external module type definition table (Column 14, lines 10-20)

logic on a computer readable medium to identify at least one support module associated with the module type in the external module type definition table,
(Column 14, lines 43-50: children corresponds to support modules)

logic on a computer readable medium for Identifying at least one module type not previously defined in the external module type definition table (Column 21, lines 30-36)

26. Berg does not teach logic on a computer readable medium to load the module based upon the module type and the at least one support module associated with the module type. However, Baker teaches logic on a computer readable medium to load the

module based upon the module type and the at least one support module associated with the module type (Pg 414: Load-order dependency may be established. Thus in a series of three drivers, which corresponds to modules, that need to be loaded, the third driver may specify all the modules it depends on, which corresponds to the support modules, for them to be loaded.)

27. It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to have modified the invention of Berg, with: logic on a computer readable medium to load the module based upon the module type and the at least one support module associated with the module type, because it allows the user to create load-order dependency.

28. As per claim 27, Berg teaches

Logic on a computer readable medium to dynamically define the at least one module type (Column 21, lines 34-36)

Logic on a computer readable medium to dynamically update the external module type definition table with the dynamically defined at least one module type (Column 14, lines 9-20; Column 14, lines 43-59)

29. As per claim 28, Berg teaches the logic to dynamically define the at least one external module type includes receiving an operator identified module type. (Column 14,

lines 55-60: the administrators corresponds to the operators. They are allowed to add new devices.)

30. As per claim 29, Berg teaches the logic to dynamically define the at least one external module type includes receiving at least one identified support modules from an operator. (Column 14, lines 43 to 60: In the case that the administrator adds a device that is classified as a children, it would corresponds to a support module.)

31. As per claim 4, it contains all the method steps that may be performed by the system of claims 11 and 14. Since claims 11 and 14 are rejected, claim 4 is rejected as well with the same motivation.

32. As per claim 7, it contains all the method steps that may be performed by the system of claims 11 and 17. Since claims 11 and 17 are rejected, claim 7 is rejected as well with the same motivation.

33. As per claim 22, it contains all the instructions that may be carried out by the system of claim 17. Since claim 17 is rejected, claim 22 is rejected as well.

34. As per claim 24, it contains all the instructions that may be carried out by the system of claim 14. Since claim 14 is rejected, claim 24 is rejected as well.

35. As per claim 5, it contains all the method steps that may be performed by the system of claims 11 and 15. Since claims 11 and 15 are rejected, claim 5 is rejected as well with the same motivation.

36. As per claim 6, it contains all the method steps that may be performed by the system of claims 11 and 16. Since claims 11 and 16 are rejected, claim 6 is rejected as well with the same motivation.

37. As per claim 8, it contains all the method steps that may be performed by the system of claims 11 and 18. Since claims 11 and 18 are rejected, claim 8 is rejected as well.

38. As per claim 23, it contains all the instructions that may be carried out by the system of claim 18. Since claim 18 is rejected, claim 23 is rejected as well.

39. As per claim 25, it contains all the instructions that may be carried out by the system of claim 16. Since claim 16 is rejected, claim 25 is rejected as well.

40. Claims 12, 13, 32 and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Berg et al., Patent No. 6449660 (Hereafter Berg) in view of: The

Windows NT Device Driver Book by Art Baker, 1997 (hereafter Baker) further in view of Managing and Developing Dynamically Loadable Kernel Modules (hereafter HP), Copyright 2001, Hewlett-Packard Company.

41. As per claim 12, Berg in view of Baker teaches the invention substantially as claimed including all of claim 11.

42. Berg in view of Baker does not teach receiving an operator generated dynamically loadable kernel module ("DLKM") type identifier. However, HP teaches a demand load DLKM for the purpose of providing a user level request for a specific module to be loaded (Chapter 12, page 501: *The Examiner notes that a DLKM is a very specific type of module and every module needs a type identifier associated with it. So in creating the DLKM, a module type definition must be coupled to the DLKM. Since the status is demand load, the Examiner considers this to be the equivalent of operator generated type identifier.*)

43. It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention to have modified the invention of Berg in view of Baker with receiving an operator generated dynamically loadable kernel module ("DLKM") type identifier, as taught by HP (Chapter 12, page 501)

44. As per claim 13, Berg does not teach receiving a computer generated dynamically loadable kernel module type identifier.

45. However, HP teaches an autoload DLKM (*Chapter 12, page 502*); (*The Examiner notes that a DLKM is a very specific type of module and every module needs a type identifier associated with it. So in creating the DLKM, a module type definition must be coupled to the DLKM. Since the status is autoload, the Examiner considers this to be the equivalent of computer generated type identifier.*) Refer to claim 12 for the motivation to combine.

46. As per claim 32, since claim 12 is rejected, and with the same motivation to combine the teachings of Berg in view of Baker and HP, claim 32 is rejected as well. (the operator generated module corresponds to operator generated DLKM)

47. As per claim 33, since claim 13 is rejected, and with the same motivation to combine the teachings of Berg in view of Baker and HP, claim 33 is rejected as well. (the operator generated module corresponds to computer generated DLKM)

48. Claims 2 and 3 are rejected under 35 U.S.C. 103(a) as being unpatentable over Berg et al., Patent No. 6449660 (Hereafter Berg) in view of Managing and Developing

Dynamically Loadable Kernel Modules (hereafter HP), Copyright 2001, Hewlett-Packard Company.

49. As per claim 2, Berg teaches the invention substantially as claimed including all of claim 1.

50. Berg does not teach receiving an operator generated dynamically loadable kernel module (“DLKM”) type identifier. However, HP teaches a demand load DLKM for the purpose of providing a user level request for a specific module to be loaded (Chapter 12, page 501: *The Examiner notes that a DLKM is a very specific type of module and every module needs a type identifier associated with it. So in creating the DLKM, a module type definition must be coupled to the DLKM. Since the status is demand load, the Examiner considers this to be the equivalent of operator generated type identifier.*)

51. It would have been obvious to one having ordinary skill in the art at the time of the applicant’s invention to have modified the invention of Berg in view of HP with receiving an operator generated dynamically loadable kernel module (“DLKM”) type identifier, as taught by HP (Chapter 12, page 501)

52. As per claim 3, Berg does not teach receiving a computer generated dynamically loadable kernel module type identifier.

53. However, HP teaches an autoload DLKM (*Chapter 12, page 502*); (*The Examiner notes that a DLKM is a very specific type of module and every module needs a type identifier associated with it. So in creating the DLKM, a module type definition must be coupled to the DLKM. Since the status is autoload, the Examiner considers this to be the equivalent of computer generated type identifier.*) Refer to claim 2 for the motivation to combine.

Response to Arguments

54. Since Naylor is no longer being used as a reference, all arguments directed to the teachings in Naylor are moot.

Conclusion

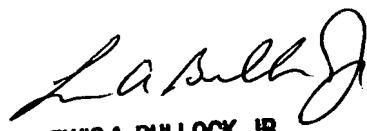
55. Relevant art not used as references above:

Forin et al., Patent No. 7,143,421

56. Applicants' amendments necessitated the new grounds of rejection presented in this office action. Accordingly, **THIS ACTION IS MADE NON-FINAL**. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MengYao Zhe whose telephone number is 571-272-6946. The examiner can normally be reached on Monday Through Friday, 10:00 - 8:00 EST. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached at 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.



LEWIS A. BULLOCK, JR.
PRIMARY EXAMINER